What’s all the fuzz about?

*Project Robus*

*Aegis™ Platform*

Adam Crain, Automatak
Chris Sistrunk PE, Mandiant
Project Robus

- Started in April 2013
- 17 advisories / 31 tickets
- Mostly DNP3, 1 Modbus
- Only 4 products so far without a detectable issue

www.automatak.com/robus
www.automatak.com/aegis
# Vendor Response Matrix

<table>
<thead>
<tr>
<th>#</th>
<th>ICS-CERT Adv</th>
<th>Company</th>
<th>Bug</th>
<th>Fix</th>
<th>Days</th>
<th>Advisory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICSA-13-161-01</td>
<td>IOServer</td>
<td>4/24</td>
<td>5/24</td>
<td>30</td>
<td>6/10/2013</td>
</tr>
<tr>
<td>2</td>
<td>ICSA-13-213-03</td>
<td>IOServer</td>
<td>5/1</td>
<td>7/20</td>
<td>80</td>
<td>8/1/2013</td>
</tr>
<tr>
<td>3</td>
<td>ICSA-13-219-01</td>
<td>SEL</td>
<td>5/1</td>
<td>5/30</td>
<td>29</td>
<td>8/7/2013</td>
</tr>
<tr>
<td>4</td>
<td>ICSA-13-226-01</td>
<td>Kepware</td>
<td>4/24</td>
<td>6/18</td>
<td>55</td>
<td>8/14/2013</td>
</tr>
<tr>
<td>8</td>
<td>ICSA-13-252-01</td>
<td>Subnet</td>
<td>4/24</td>
<td>8/30</td>
<td>128</td>
<td>9/9/2013</td>
</tr>
<tr>
<td>10</td>
<td>ICSA-13-297-01</td>
<td>Catapult</td>
<td>4/24</td>
<td>10/1</td>
<td>160</td>
<td>11/22/2013</td>
</tr>
<tr>
<td>11</td>
<td>ICSA-13-297-02</td>
<td>GE IP S.R.</td>
<td>S.R.</td>
<td>10/1</td>
<td>n/a</td>
<td>11/22/2013</td>
</tr>
<tr>
<td>13</td>
<td>ICSA-13-346-02</td>
<td>Cooper OPC</td>
<td>7/31</td>
<td>None</td>
<td>∞day™</td>
<td>12/12/2013</td>
</tr>
<tr>
<td>14</td>
<td>ICSA-13-346-01</td>
<td>Cooper/Cybectec</td>
<td>5/1</td>
<td>12/12</td>
<td>225</td>
<td>12/12/2013</td>
</tr>
<tr>
<td>15</td>
<td>ICSA-13-352-01</td>
<td>Novatech</td>
<td>5/1</td>
<td>9/5</td>
<td>127</td>
<td>12/18/2013</td>
</tr>
<tr>
<td>16</td>
<td>ICSA-14-014-01</td>
<td>Schneider</td>
<td>8/6</td>
<td>8/23</td>
<td>17</td>
<td>1/14/2014</td>
</tr>
<tr>
<td>17</td>
<td>ICSA-14-006-01</td>
<td>Schneider/Telvent</td>
<td>8/29</td>
<td>10/16</td>
<td>48</td>
<td>1/30/2014</td>
</tr>
</tbody>
</table>
Breaking Down DNP3

Ref from IEEE Std 1815-2012

TCP 20000
TCP 19999 (TLS)
UDP 20000
White Noise Fuzzing

#1 random == really “dumb”
Template (mutational) Fuzzing
Generational “Smart” Fuzzing
Multi-field Anomalies
Generational == most vulns!
Comparing template and generational fuzzing

- Time: 118
- Test Cases: 10,000,000
- Bugs Found: 8

Legend:
- Template
- Overlap
- Generational

3/13/14 | Copyright © Codenomicon, 2001 - 2014. All rights reserved. | 1
Aegis™ Specifics

• Written in Scala [www.scala-lang.org](http://www.scala-lang.org)
• Currently porting it to C#
• Protocol boundary conditions
• Abstracts physical layer
• Combines aspects of generation and mutation
• Repeatable random seeds
• ~200,000 test cases with one seed
Fuzzer Test Flow

- x Num Test Cases
- Test DNP3 Message (DL, TL, or AL)
- Request Link States
  - Request
  - Response
  - Link Status
- x Num Retry (10)
Combinatorics

```scala
val nums = List(1, 3)
val colors = List("red", "green")

// repeat the reversed string num times
def combine(i: Int, s: String) = List.fill(i)(s.reverse).mkString

val result = Cartesian.Transform(colors, nums)(combine)
```

What is result?
Lazy Generator

// val nums = List(1, 3)
// val colors = List("red","green")
> result.foreach(println)
der
derderder
neerg
neergneergneerg
Fuzzing is $O(2^n)$

\[
\{ \text{frames} \} = f(\text{byte, Type})
\]

\[
\begin{align*}
\{ \text{byte} \} &= f(\text{bool, bool, int}) \\
\{ \text{true, false} \} \\
\{ \text{true, false} \} \\
\{ 0, 1, 63 \} \\
\end{align*}
\]

\[
\{ \text{Type} \} = f(\ldots)
\]

{ frames } = f (byte, Type)
Generators can get large!

- Many function codes
- Many objects
- Header types
- Many field values
Types of Vulnerabilities
Using Aegis

Aegis Platform - CONFIDENTIAL - Automatak, LLC

Required argument not found: pid (Procedure id within module)

usage: aegis-console [flags ...]

Valid module ids: [dn3]

- mid <arg>    Module id of protocol
- pid <arg>    Procedure id within module
- host <arg>(127.0.0.1) IP address for client connection
- port <arg>(20000)[0, 65535] Port to connect or listen on
- listen      Listens on the specified port instead of connecting
- help        Prints help information
- start <arg>  Starts testing at a specified test case #
- count <arg>  Limits execution to the specified number of test cases
- repeats <arg>(1)[min=1] Number of times to repeat the specified test case

SANS
So easy…Twitter can do it.
Examples

Run 10 link layer test cases starting at #123

$ aegis-console -mid dnp3 -pid lfuzz -start 123 -count 10

Unsolicited response fuzzing of a master listening on default port 20000 with master address of 0 and an outstation address of 1

$ aegis-console -mid dnp3 -pid aufuzz -dest 0 -src 1 -master -listen

Outstation link layer fuzzing test case #100 only

$ aegis-console -mid dnp3 -pid lfuzz -start 100 -count 1

Outstation application object fuzzing against 192.168.1.55:20001 with default addressing

$ aegis-console -mid dnp3 -id aofuzz -host 192.168.1.55 -port 20001
Recorded Demos

Video 1: a DNP3 outstation
-pid aofuzz

Video 2: a DNP3 master
-pid aufuzz -listen -master -seed 1
White-box vs. Black-box Testing

• Defender has the advantage, but has to choose to exercise it.
• Software-based solutions allow developers to test continually.

There are many OSS tools of the trade.
Code Coverage with gcov

• If you don’t run a line of code, you’ll never find a bug in it
• Important metric, but not a guarantee of success
Dynamic Analysis with Valgrind

- Virtualized binary execution
  - hooks system calls
- memcheck is your friend
  - leaks
  - overrun / underrun
  - user after free
- callgrind
  - find true bottlenecks

valgrind.org
Wurldtech Achilles - openDNP3 outstation - SELECT code coverage
Mu4000 - open dnp3 outstation - SELECT code coverage
Aegis™ 0.1.0 - open dnp3 outstation - SELECT code coverage
8%

33% more relative to others
Total code coverage
More Fuzzers are Better

- Aegis fuzzing with every release
- In-memory fuzzing
- Checks from Codenomicon, Wurldtech, Mu

open dnp3
Fuzzing is just another tool

- Unit test coverage in excess of 90%
- Valgrind for dynamic analysis
- Open source conformance test harness
- Static analysis using Clang / Coverity / CppCheck
Security!

HOST Grant

- Adding authentication (SAv5)
- Adding encryption (TLS)

http://investments.opencybersecurity.org/
SHODAN update

Probably default configs

• Many similar responses
• Same DNP Addresses

_python shell_

```python
>>> " ".join("%02x" % ord(i) for i in "DNP3 paste from shodan")
```

Unsolicited Response, IIN
Restart & Need Time Synch

Unsolicited Response with Binary and Analog Data
Class 1/2/3/0 Poll!!!
Conclusions

• DNP3 is not a special case, other protocols same fate
  Modbus, IEC 60870, IEC 61850, ICCP, EtherNet/IP…
• Early testing both slave/server AND master/client sides of the protocol are important!
• Compliance != Security, but the culture is important
• Don’t have to be a nation/state or large firm to do this
• A few good folks can make a difference in the industry
Go fuzz yourself before it does...

Questions?

@jadamcrain
@chrissistrunk